☆ Build the Subsequences

A subsequence of string, *s*, is obtained by deleting one or more characters from *s*. For example, the set of subsequences for string *s = "abc"* would be *{"a", "ab", "ac", "abc", "b", "bc", "c", ""}* (recall that the empty string is a subsequence of all strings).

Complete the *buildSubsequences* function in your editor. It has *1* parameter: a string, *s*. It must return an array of strings containing all possible subsequences of *s* in *alphabetical* order. *Do not* include the empty string in your returned array of subsequences.

Input Format

The locked stub code in your editor reads a single string, *s*, from stdin and passes it to your function.

Constraints

- *1 < |s| < 16*
- *s* is a string of unique lowercase letters *(a-z)*.

Output Format

Output to stdout is handled by the locked stub code in your editor, which prints each element of the returned array on a new line.

Sample Input 1

```
ba
```

Sample Output 1

```
a
b
ba
```

Sample Input 2

```
abc
```

Sample Output 2

```
a
ab
abc
ac
b
bc
c
```

Explanation

Sample Case 1

s = ba

There are *3* subsequences *b, a, ba*

Sample Case 2

s = abc

There are *7* subsequences *a, ab, abc, ac, b, bc, c*

YOUR ANSWER

We recommend you take a quick tour of our editor before you proceed. The timer will pause up to 90 seconds for the tour.

Start tour

Original code | Java 7

```
1 ▸ import ↔;
6
7   public class Solution {
8
```

```
/*
 * Complete the function below.
 */

    static String[] buildSubsequences(String s) {
        if (s == null || s.length() == 0) {
            return null;
        }

        LinkedHashSet<String> set = new LinkedHashSet<String>
();
        dfs(set, 0, new StringBuilder(), s);

        String[] res = new String[set.size()];
        int i = 0;
        for (String sub : set) {
            res[i] = sub;
            i++;
        }

```

Fall 2016 - CMU OCI

25m : 47s
to test end

```
    }

    static void dfs(LinkedHashSet<String> set, int start,
StringBuilder tmp, String s) {
        if (start == s.length()) {
            String sub = tmp.toString();
            if (sub.length() > 0) {
               set.add(sub);
            }
            return;
        }

        dfs(set, start + 1, tmp, s);
        tmp.append(s.charAt(start));
        dfs(set, start + 1, tmp, s);
        tmp.deleteCharAt(tmp.length() - 1);

        return;
    }

```

```
52 ▶    public static void main(String[] args) throws IOException{
↔}
72  }
```

Line: 24 Col: 13

Test against custom input

Run Code | Submit code & Continue

(You can submit any number of times)

Download sample test cases *The input/output files have Unix line endings. Do not use Notepad to edit them on windows.*

Compiled successfully. All available test cases passed!

Test Case #1: ✔ Test Case #5: ✔

Fall 2016 - CMU OCI

25m : 47s to test end

Test Case #4: ✔ Test Case #8: ✔

Testcase 1: *Success*

Your Output

```
a
b
ba
```

Expected Output

```
a
b
ba
```

Testcase 2: *Success*

Your Output

```
a
ab
abc
ac
```

```
b
bc
c
```

Expected Output

```
a
ab
abc
ac
b
bc
c
```

Testcase 3: *Success*

Your Output

```
Output hidden
```



```
Output hidden
```

Testcase 5: *Success*

Your Output

```
Output hidden
```

Testcase 6: *Success*

Your Output

```
Output hidden
```

Testcase 7: *Success*

Your Output

```
Output hidden
```

Testcase 8: *Success*

Your Output

```
Output hidden
```


Fall 2016 - CMU OCI

25m : 47s
to test end

1

2

3

4

5